Dataset Distillation via Factorization
–*Supplementary Materials*–

Songhua Liu **Kai Wang** **Xingyi Yang** **Jingwen Ye** **Xinchao Wang**
National University of Singapore
{songhua.liu,e0823044,xyang}@u.nus.edu, {jingweny,xinchao}@nus.edu.sg

Appendices

In this part, we provide additional details, more results, potential limitations, and future directions of the proposed Hallucinator-Basis factorization (HaBa) for dataset disllation (DD). First, we provide more details on the pipeline of HaBa. Then, we conduct more experiments to demonstrate and analyze performance of our method, including results on more benchmarks with larger resolutions, as supplement to the quantitative study in the main paper. We also provide more qualitative results by HaBa and additional ablation studies. Finally, we discuss some limitations and future works of our method.

Algorithm 1 Hallucinator-Basis Factorization (HaBa) for Dataset Distillation.

Input: $\mathcal{T}$: original dataset; $|\mathcal{H}|$: total number of hallucinators; $|\mathcal{B}|$: total number of bases; η_H: learning rate of hallucinators; η_B: learning rate of bases; η_F: learning rate of feature extractor.
Output: $\mathcal{H}$: a set of hallucinators; $\mathcal{B}$: a set of bases;

Randomly initialize hallucinators $\mathcal{H} = \{H_{\theta_j}\}_{j=1}^{|\mathcal{H}|}$, bases $\mathcal{B} = \{(\hat{x}_i, \hat{y}_i)\}_{i=1}^{|\mathcal{B}|}$, and parameters ψ of the feature extractor F;
while not done **do**
  $\mathcal{H}' \leftarrow$ a random batch of hallucinators from $\mathcal{H}$;
  $\mathcal{B}' \leftarrow$ a random batch of bases from $\mathcal{B}$;
  for each $1 \leq i \leq |\mathcal{B}'|$ **do**
    for each $1 \leq j \leq |\mathcal{H}'|$ **do**
      $\tilde{x}_{ij} = H_{\theta_j}(\hat{x}_i)$;
    end for
  end for
  Compute $\mathcal{L}_{\mathcal{S}}$ using Eq. 6 of the main paper;
  Update $\mathcal{H}$: $\theta_i \leftarrow \theta_i - \eta_H \nabla_{\theta_i} \mathcal{L}_{\mathcal{S}}$ for $1 \leq j \leq |\mathcal{H}'|$;
  Update $\mathcal{B}$: $\hat{x}_i \leftarrow \hat{x}_i - \eta_B \nabla_{\hat{x}_i} \mathcal{L}_{\mathcal{S}}$ and $\hat{y}_i \leftarrow \hat{y}_i - \eta_B \nabla_{\hat{y}_i} \mathcal{L}_{\mathcal{S}}$ (optional) for $1 \leq i \leq |\mathcal{B}'|$;
  Compute $\mathcal{L}_F$ using Eq. 4 of the main paper;
  Update F: $\psi \leftarrow \psi - \eta_F \nabla_{\psi} \mathcal{L}_F$;
end while

Appendix A Algorithm Details

To better elaborate the details of the proposed HaBa for DD, we provide an algorithmic illustration for the whole pipeline in Alg. 1, as a supplement to Sec. 3 of the main paper. The overall algorithm takes an original dataset as well as some hyper-parameters shown in Alg. 1 as input. The output is the distilled result including a set of hallucinators $\mathcal{H}$ and a set of bases $\mathcal{B}$, as defined in Eq. 1 of the

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

	Dataset	MNIST			FashionMNIST		
	IPC	1	10	50	1	10	50
	Ratio %	0.017	0.17	0.83	0.017	0.17	0.83
Coreset	Random	64.9±3.5	95.1±0.9	97.9±0.2	51.4±3.8	73.8±0.7	82.5±0.7
	Herding	89.2±1.6	93.7±0.3	94.8±0.2	67.0±1.9	71.1±0.7	71.9±0.8
	K-Center	89.3±1.5	84.4±1.7	97.4±0.3	66.9±1.8	54.7±1.5	68.3±0.8
	Forgetting	35.5±5.6	68.1±3.3	88.2±1.2	42.0±5.5	53.9±2.0	55.0±1.1
Distillation	DD [10]	-	79.5±8.1	-	-	-	-
	LD [2]	60.9±3.2	87.3±0.7	93.3±0.3	-	-	-
	DC [15]	91.7±0.5	97.4±0.2	98.8±0.2	70.5±0.6	82.3±0.4	83.6±0.4
	DSA [13]	88.7±0.6	**97.8±0.1**	**99.2±0.1**	70.6±0.6	84.6±0.3	88.7±0.2
	DM [14]	89.7±0.6	97.5±0.1	98.6±0.1	-	-	-
	CAFE [9]	**93.1±0.3**	97.2±0.2	98.6±0.2	77.1±0.9	83.0±0.4	84.8±0.4
	CAFE+DSA [9]	90.8±0.5	97.5±0.1	98.9±0.2	73.7±0.7	83.0±0.3	88.2±0.3
	MTT [3]	88.7±1.0	96.6±0.4	98.1±0.1	75.7±1.5	88.4±0.4	90.0±0.1
Factorization	BPC	1	9	49	1	9	49
	Ratio %	0.034	0.17	0.83	0.034	0.17	0.83
	HaBa	92.4±0.4	97.4±0.2	98.1±0.1	**80.9±0.7**	**88.6±0.2**	**90.3±0.1**
Whole Dataset		99.6±0.0			93.5±0.1		

Table 1: The performance (test accuracy %) comparison with state-of-the-art methods on MNIST and FashionMNIST datasets. IPC: Number of Images Per Class; BPC: Number of Bases Per Class; Ratio (%): the ratio of distilled images to the whole training set.

Depth	0	1	2	3
Accuracy (%)	68.43±0.37	70.27±0.63	71.17±0.29	71.55±0.27
Downstream Speed	144.54	140.11	125.04	115.62
# of Parameters	6,144	6,312	10,963	16,131

Table 3: Ablation studies on the depth (number of nonlinear blocks) of hallucinator.

# of Channels	3	8	16
Accuracy	70.27±0.63	70.47±0.37	71.28±0.35
Downstream Speed	140.11	138.48	135.12
# of Parameters	6,312	16,827	33,651

Table 4: Ablation studies on the number of feature channels in hallucinator.

main paper. The goal is to equip the distilled dataset with similar downstream performance to the original one.

Appendix B More Results

Low-Resolution Data: We provide results on the more-common benchmark datasets in DD in Tab. 1: MNIST [7] and FashionMNIST [12]. Both datasets contain 60,000 images for training and 10,000 images for testing in 10 classes. The images are under 28×28 resolution with 1 channel. We build our HaBa on MTT [3] in this part. Although the performances of DD on these two dataset seem to be saturated, our method may still yield consistent improvement over the baseline, especially when the ratio of distilled images to the whole training set is small.

ImageNet Subsets: We also evaluate the proposed scheme on the more-challenging settings of ImageNet [4] subsets. We follow the baseline MTT [3] for the divisions of subsets. The 6 subsets include ImageFruit, ImageMeow, ImageNette, ImageSquawk, ImageWoof, and ImageYellow. Each subset contains over 10,000 images, and we resize all the images to 128×128 resolution following the original setting. We use ConvNet with 5 `Conv-InstanceNorm-ReLU-AvgPool` layers for training. For testing, in addition to the same structure of ConvNet, we also evaluate the results under 3 other architectures: ResNet, VGG, and AlexNet. To ensure the same number of parameters used for the distilled datasets, we set the number of images per class used by the baseline as the number of bases per class used by HaBa plus 1, *i.e.*, 2 IPC v.s. 1 BPC and 11 IPC v.s. 10 BPC. Other settings follow the same configuration in the main paper.

The test performances of models trained by the distilled datasets are shown in Tab. 2. We can observe that HaBa outperforms the baseline in almost all cases except several experiments when IPC and BPC are small and the architectures of training and testing are the same. Notably, in all the cross-architecture generalization settings, HaBa achieves superior performance over the baseline, which further demonstrates the improvement of data efficiency introduced by the factorization and online pair-wise combination.

	Method	ConvNet		ResNet		VGG		AlexNet	
	IPC BPC	2 1	11 10	2 1	11 10	2 1	11 10	2 1	11 10
ImageFruit	Baseline	31.76±1.64	40.12±1.87	24.36±2.20	31.24±1.71	30.20±1.43	42.52±1.16	27.92±1.84	29.88±1.60
	w. HaBa	34.68±1.13	42.52±1.56	26.60±2.48	33.08±1.02	31.92±1.91	45.12±1.18	28.16±1.29	32.84±1.69
	Gain	+2.92	+2.40	+2.24	+1.84	+1.72	+2.60	+0.24	+2.96
ImageMeow	Baseline	35.28±2.23	41.00±1.45	17.64±1.51	19.64±0.93	31.52±1.27	39.44±1.23	21.04±1.64	22.04±1.72
	w. HaBa	36.92±0.93	42.92±0.86	25.44±1.02	26.28±2.61	35.00±0.76	47.68±0.57	23.76±2.06	24.04±1.94
	Gain	+1.64	+1.92	+7.80	+6.64	+3.48	+8.24	+2.72	+2.00
ImageNette	Baseline	55.16±1.08	63.88±0.48	25.52±1.31	42.80±1.49	47.48±1.67	62.80±1.59	30.96±0.97	34.60±2.95
	w. HaBa	51.92±1.65	64.72±1.60	28.88±2.61	46.84±1.25	47.80±1.21	63.76±1.05	33.28±1.98	40.84±1.80
	Gain	-3.24	+0.84	+3.36	+4.04	+0.32	+0.96	+2.68	+6.24
ImageSquawk	Baseline	43.92±0.63	54.64±0.96	30.64±1.47	46.40±1.85	39.36±1.83	52.00±1.91	22.04±1.80	34.20±2.08
	w. HaBa	41.88±1.37	56.80±1.04	31.52±2.39	48.92±1.77	39.64±1.78	56.88±0.84	23.28±0.55	35.00±1.72
	Gain	-2.04	+2.16	+0.88	+2.52	+0.28	+4.88	+1.24	+0.80
ImageWoof	Baseline	30.92±1.26	36.56±0.75	16.24±1.48	18.12±0.47	25.60±0.69	29.36±1.23	22.68±1.42	23.68±1.37
	w. HaBa	32.40±0.67	38.60±1.26	20.20±1.55	25.20±0.95	27.08±1.81	37.44±1.08	24.88±1.20	27.72±1.12
	Gain	+1.48	+2.04	+3.96	+7.08	+1.48	+8.08	+2.20	+4.04
ImageYellow	Baseline	49.72±1.38	60.40±1.46	29.08±1.99	42.72±1.24	44.04±1.46	50.84±0.56	28.60±1.48	35.60±2.03
	w. HaBa	50.44±1.56	63.00±1.61	36.32±0.65	48.48±1.55	47.28±1.59	57.24±1.01	29.08±1.19	36.44±1.21
	Gain	+0.72	+2.60	+7.24	+5.76	+3.24	+6.40	+0.48	+0.84

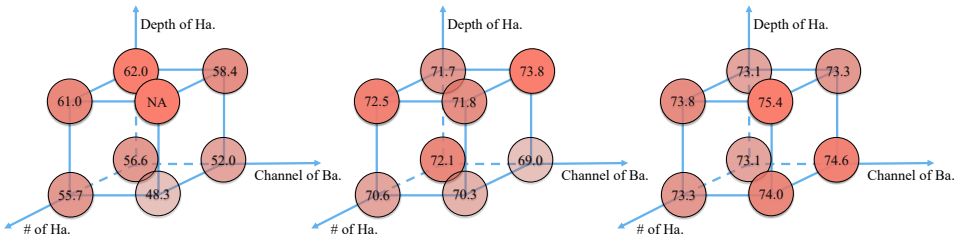
Table 2: Cross-architecture performance (test accuracy %) comparison with the baseline on various subsets of ImageNet dataset.

BPC	1	10	50
Ours	55.66±0.29	70.27±0.63	74.04±0.16
Share Enc. & Dec.	55.14±0.44	69.47±0.09	72.69±0.39
Baseline	49.89±0.95	65.92±0.62	70.73±0.52

Table 5: Impact on sharing encoder and decoder across all hallucinators.

SPC	Rand	Herd	DSA	DM	IDC	IDC w. HaBa	Whole Dataset
10	42.6	56.2	65.0	69.1	73.3	74.5	93.4
20	57.0	72.9	74.0	77.2	83.0	84.3	

Table 6: Results of speech recognition on Mini Speech Commands.



(a). Small Storage Budget
10×2×32×32×3 Param.
Baseline Acc.: 49.9%

(b). Medium Storage Budget
10×11×32×32×3 Param.
Baseline Acc.: 65.9%

(c). Large Storage Budget
10×2×51×32×3 Param.
Baseline Acc.: 70.7%

Figure 1: Exploration on the configurations of different factors in hallucinators and bases.

More Ablations on Hallucinators: In the default setting, the encoder and decoder of hallucinators have 1 `Conv-ReLU` block and the number of feature channels is 3. In this part, we provide more results when we consider increasing the capacity of the hallucination networks. As shown in Tab. 3, we try increasing the depth of the networks by adding more nonlinear blocks. Although the performance can indeed be improved, it results in nonnegligible latency to downstream training speed, measured by the number of epochs per second. Taking both training speed and performance into consideration, we consider using only 1 nonlinear block by default, which yields best trade-off between the two factors. Likewise, we also try increasing the number of feature channels in halluciantors as shown in Tab. 4. The number of parameters is almost proportional to the number of channels. However, the performance gain is very limited. Thus, we simply take the number of channels in images, which is 3 for RGB images, as the number of feature channels in hallucinators.

More Insights on the Configurations of Hallucinators and Bases: As shown in the ablation studies in the main paper and the supplement, under the framework of hallucinator-basis factorization, there are many factors that affect the performance. Given a fixed storage budget, how to scale the bases and hallucinators is an important topic. Among all the factors, we empirically find that the depth of hallucinators, the number of hallucinators, the number of channels in each basis, and the number of bases are the most important ones, which will be studied in the following exploration. Here, we

consider three types of storage budget: small, medium, and large, corresponding to the cost of IPC=2, 11, and 51 for the baseline method respectively. We consider cases of 1 and 2 convolution blocks for the depth of hallucinators, 2 and 5 for the number of hallucinators, and 1 and 3 for the number of channels in each basis. For each setting, we adjust the number of bases to fit the given budget. Enumerating all the configurations, there are totally 8 settings for each kind of budget. Their results are visualized in Fig. 1. Based on the results, we have the following observations:

- **For all the three types of budget, the best performance is achieved by using deeper hallucinators. Especially under small and medium budgets, using depth 2 can outperform using depth 1 almost consistently.** This can be explained by the more complex sample-wise relationship extracted by hallucinators.
- In our framework, bases are expected to store sample-independent information while hallucinators are used to encode shared relationship across all the samples. **When the budget is small, using 1-channel bases can achieve significantly better results.** This is because small storage budget would more rely on increasing the number of independent data samples for a better diversity. The informativeness of each basis appears less important.
- **When the budget increases, the advantage of 1-channel bases mentioned before would diminish gradually. Especially under the large budgets, 3-channel bases outperform 1-channel ones consistently.** The reason is that when the number of bases is adequate, focusing on the informativeness of each basis can produce more benefit than increasing the number.
- **When the budget is large, using more hallucinators can yield slightly better results**, which can probably be attributed to the further improvement on the diversity.
- **The larger the budget is, the less insensitive the performance is, to different configurations.**

Note that the above exploration is conducted without taking the downstream training speed into consideration, which is also an important metric in the task of dataset distillation. Our opinion on the scalability is that, when downstream training overhead is not a issue, deeper hallucinators are recommended for better performance; otherwise if downstream efficiency is desired, we find that 1 nonlinear block is sufficient, since heavier hallucination networks can result in nonnegligible latency, especially when the total number of images is large.

Sharing Encoder and Decoder across all Hallucinators: As a variant of our default case which uses different hallucination networks, it is also feasible for the halluciantors to share a common group of encoder and decoder but use different parameters (σ, μ) for affine transformation, which is potential to further boost the data efficiency. As shown in Tab. 5, the performance becomes slightly worse. We conjecture that different convolution encoders and decoders may contribute to the diversity of the extracted patterns, which increases the representation ability of the hallucinator set. Moreover, since we only use 1 convolution block for encoders and decoders, the number of parameters is not so significant compared with that of a basis. Therefore, we consider making the whole network independent with each other for all hallucinators.

Results on Speech Domain: To validate the versatility of the proposed hallucinator-basis factorization solution, we further conduct experiments on the speech domain using Mini Speech Commands [11], which contains 8,000 audio clips for 8 command classes. We adopt IDC [6] as the baseline and all the protocols for comparisons follow the official settings. We compare our method with the coreset selection based Random and Herding, DSA [13], DM [14], and the IDC baseline [6]. The results in Tab. 6 shows that our method can produce consistent improvement on the downstream test accuracy, which further reflects the generality of our method for different modalities. Here, SPC denotes the number of speech spectrograms per class.

Robustness to Corruption: We further examine the generalization performance of our method and the baseline one on CIFAR10-C [5], the corrupted version of CIFAR10 dataset with 19 different types of corruption. There are five corrupted levels from 1 (mildest) to 5 (severest) and we report the mean test accuracy across 19 domains on different levels. Since the proposed method can increase the accuracy and alleviate the under-fitting problem on the original domain, which is one dominant component of cross-domain generalization [1], it can also demonstrate superior robustness in all corrupted data as demonstrated in Fig. 2. Also, the gap between performance using distilled dataset and original dataset becomes smaller with the increase of corrupted level, which suggests that our

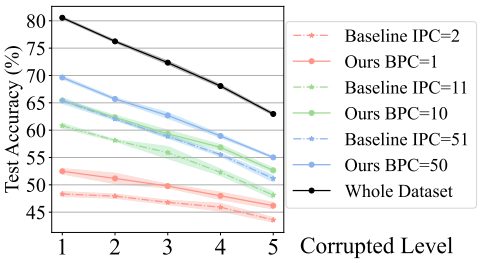
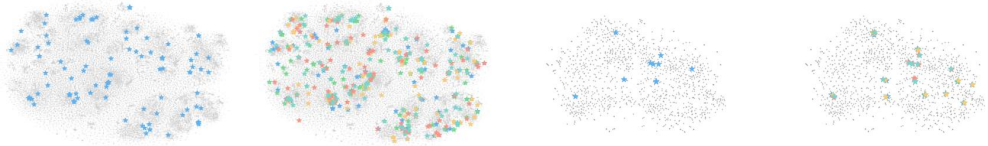


Figure 2: Generalization performance on images with different corrupted levels.

Hyper-Parameter	Notation	Value
Height of Basis	h'	Height of Image h
Width of Basis	w'	Width of Image w
Channel of Basis	c'	Channel of Image c
Channel of Hallucinator	c''	Channel of Image c
Depth of Hallucinator	-	1
Learning Rate of Feature Extractor	η_F	0.001
Weight of $\mathcal{L}_{con.}$	$\lambda_{con.}$	0.1
Weight of $\mathcal{L}_{task}$	λ_{task}	1
Weight of $\mathcal{L}_{DD}$	λ_{DD}	1
Weight of $\mathcal{L}_{cos.}$	$\lambda_{cos.}$	0.1

Table 7: List of hyper-parameters.



(a) MTT on CIFAR10 (b) HaBa on CIFAR10 (c) MTT on ImageSquawk (d) HaBa on ImageSquawk

Figure 3: TSNE visualization of results by our HaBa and baseline MTT on CIFAR10 and ImageSquawk datasets. Markers with different colors in our results denote images generated by different hallucinators. Gray dots denote real images.

method improves the domain generalization ability potentially, thanks to the diverse training data composed of hallucinators and bases.

List of Hyper-Parameters: In Tab. 7, we provide a clear view of the hyper-parameters used in this paper. All the experiments follows these settings if not specified. The performance of our method is insensitive to the values of these hyper-parameters as analyzed in the ablation studies in both the main paper and the appendix. Other hyper-parameters not listed come from the adopted baseline methods and we follow their original settings.

TSNE Visualizations: To provide a better understanding on why the HaBa factorization can help on data efficiency in dataset distillation, we adopt TSNE [8] to visualize the features before the last linear layer of a teacher model trained on the original datasets. In Fig. 3, we plot features of both original images and the distilled ones. The results reveal that datasets restored from our hallucinators and bases can describe the original data distribution more finely, which means that the original datasets can be represented with the distilled ones with less information loss. Given that the total numbers of parameters used for storing distilled datasets are the same, our method can improve the data efficiency significantly.

Visualizations of Factorized Results: We first provide the full results of HaBa factorization on CIFAR10 dataset with 5 hallucinators and 10 BPC in Fig. 4, as a supplement to Fig. 4 in the main paper. We also provide the distilled results on datasets with larger resolutions in Fig. 5 and 6 for the above 6 ImageNet subsets. Here, we use 1 BPC and 2 hallucinators for visualization. Through these results, we can find that bases in our scheme mainly define the basic contents, while different hallucinators may transform each basis to different appearances and styles. Such difference is encouraged to be as large as possible to diversify the distilled data and thus improve data efficiency during the end-to-end training pipeline of DD.

References

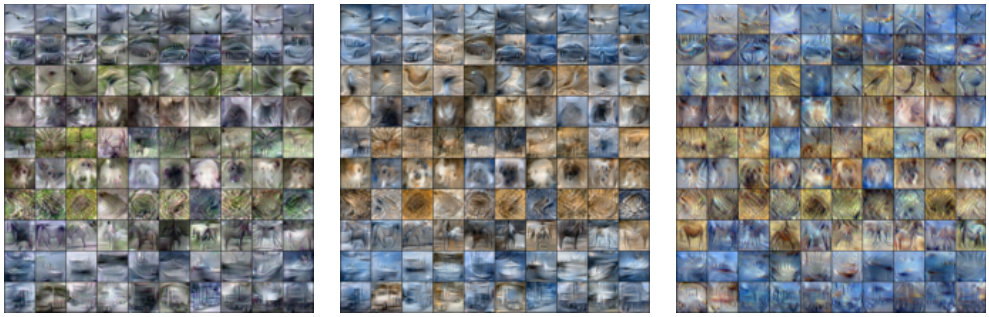
[1] Shai Ben-David, John Blitzer, Koby Crammer, Alex Kulesza, Fernando Pereira, and Jennifer Wortman Vaughan. A theory of learning from different domains. *Machine learning*, 79(1):151–175, 2010.

[2] Ondrej Bohdal, Yongxin Yang, and Timothy Hospedales. Flexible dataset distillation: Learn labels instead of images. *arXiv preprint arXiv:2006.08572*, 2020.

[3] George Cazenavette, Tongzhou Wang, Antonio Torralba, Alexei A Efros, and Jun-Yan Zhu. Dataset distillation by matching training trajectories. *arXiv preprint arXiv:2203.11932*, 2022.

[4] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

(a) Bases (b) Images by H_1 (c) Images by H_2



(d) Images by H_3 (e) Images by H_4 (f) Images by H_5

Figure 4: Visualization of factorized results by our HaBa on CIFAR10. Please zoom-in for better visualization.

[5] Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. *Proceedings of the International Conference on Learning Representations*, 2019.

[6] Jang-Hyun Kim, Jinuk Kim, Seong Joon Oh, Sangdoo Yun, Hwanjun Song, Joonhyun Jeong, Jung-Woo Ha, and Hyun Oh Song. Dataset condensation via efficient synthetic-data parameterization. *arXiv preprint arXiv:2205.14959*, 2022.

[7] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[8] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(11), 2008.

[9] Kai Wang, Bo Zhao, Xiangyu Peng, Zheng Zhu, Shuo Yang, Shuo Wang, Guan Huang, Hakan Bilen, Xinchao Wang, and Yang You. Cafe: Learning to condense dataset by aligning features. *arXiv preprint arXiv:2203.01531*, 2022.

[10] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018.

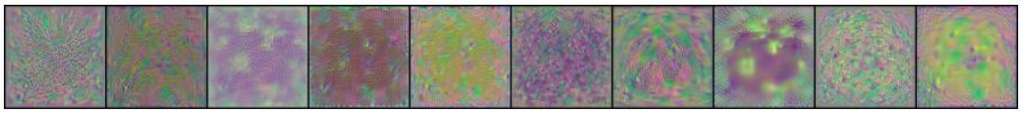
[11] Pete Warden. Speech commands: A dataset for limited-vocabulary speech recognition. *arXiv preprint arXiv:1804.03209*, 2018.

[12] Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms. *arXiv preprint arXiv:1708.07747*, 2017.

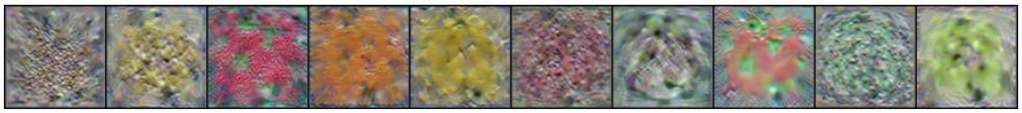
[13] Bo Zhao and Hakan Bilen. Dataset condensation with differentiable siamese augmentation. In *International Conference on Machine Learning*, pages 12674–12685. PMLR, 2021.

[14] Bo Zhao and Hakan Bilen. Dataset condensation with distribution matching. *arXiv preprint arXiv:2110.04181*, 2021.

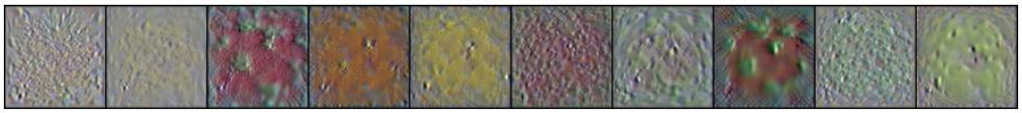
[15] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. *arXiv preprint arXiv:2006.05929*, 2020.



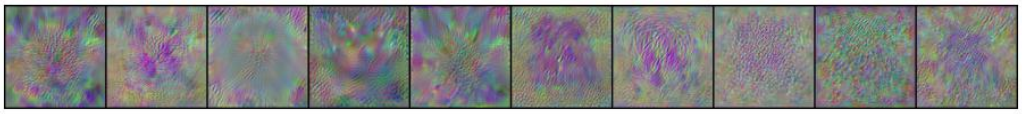
(a) Bases on ImageFruit.



(b) Images by H_1 on ImageFruit.



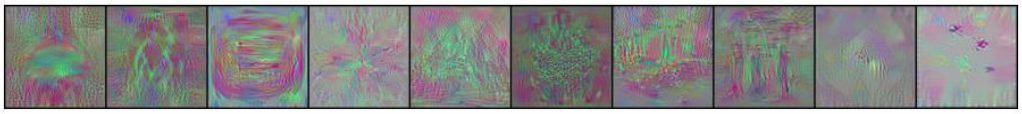
(c) Images by H_2 on ImageFruit.



(d) Bases on ImageMeow.

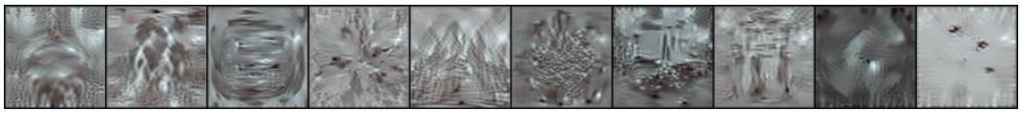
(e) Images by H_1 on ImageMeow.

(f) Images by H_2 on ImageMeow.



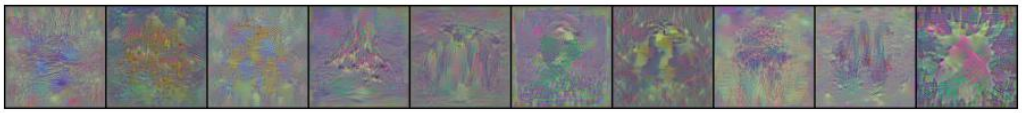
(g) Bases on ImageNette.

(h) Images by H_1 on ImageNette.



(i) Images by H_2 on ImageNette.

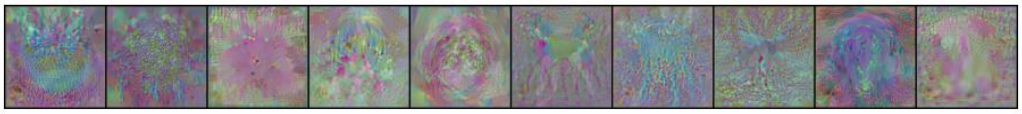
Figure 5: Visualization of factorized results by our HaBa on ImageNet subsets.



(a) Bases on ImageSquawk.

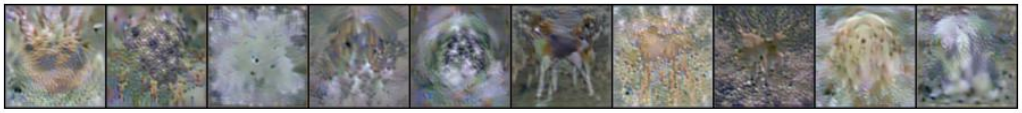
(b) Images by H_1 on ImageSquawk.

(c) Images by H_2 on ImageSquawk.

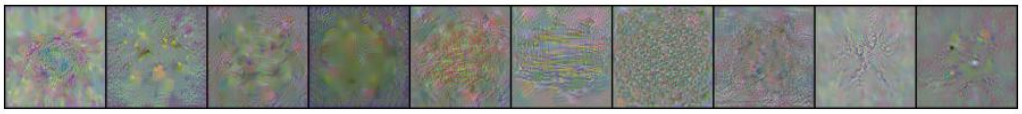


(d) Bases on ImageWoof.

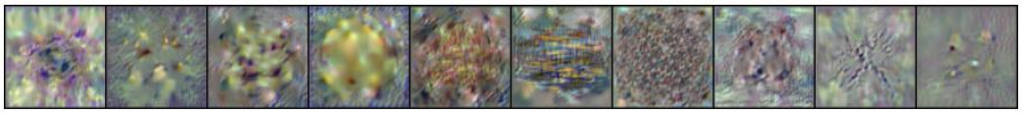
(e) Images by H_1 on ImageWoof.



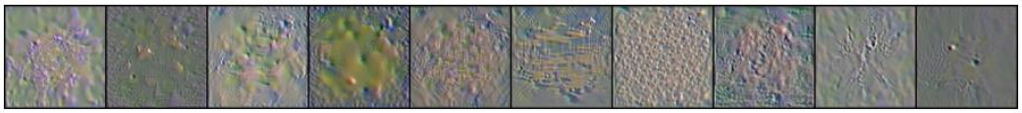
(f) Images by H_2 on ImageWoof.



(g) Bases on ImageYellow.



(h) Images by H_1 on ImageYellow.



(i) Images by H_2 on ImageYellow.

Figure 6: Visualization of factorized results by our HaBa on ImageNet subsets (Cont.).